



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

Am

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/765,882	01/16/2001	Loay Abu-Husein	10992693-1	9296

7590 06/15/2005

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort collins, CO 80527-2400

EXAMINER
----------

ZHOU, TING

ART UNIT	PAPER NUMBER
	2173

DATE MAILED: 06/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/765,882	ABU-HUSEIN, LOAY	
	<b>Examiner</b>	<b>Art Unit</b>	
	Ting Zhou	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM  
 THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### **Status**

1) Responsive to communication(s) filed on 17 May 2005.  
 2a) This action is **FINAL**.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### **Disposition of Claims**

4) Claim(s) 1,2,6-14 and 16-25 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1,2,6-14 and 16-25 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### **Application Papers**

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### **Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### **Attachment(s)**

1)  Notice of References Cited (PTO-892)  
 2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
 Paper No(s)/Mail Date \_\_\_\_\_

4)  Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_

5)  Notice of Informal Patent Application (PTO-152)  
 6)  Other: \_\_\_\_\_

**DETAILED ACTION**

1. The amendment filed on 17 May 2005 have been received and entered. Claims 1-2, 6-14 and 16-25 as amended are pending in the application.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-2, 6-14 and 16-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yan et al. U.S. Patent 6,003,065 and Inoue et al. U.S. Patent 6,456,388.

Referring to claim 1, Yan et al. teach an appliance application loading system for a network environment (Yan et al.: column 2, lines 54-60). Specifically, Yan et al. teach a client (a thin client via the peripheral device) (Yan et al.: column 19, lines 38-40); a web application server communicating with the client within the network environment and configured as a central location to store applications and configuration settings for an appliance to enable configuration setups for a plurality of appliances (a host computer communicating with the peripheral devices over the network; the host computer stores applications requiring use of peripheral devices and peripheral device profiles for automatic configuration of the peripheral device) (Yan et al.: column 6, lines 52-67, column 20, line 63 - column 21, line 37 and column 23, lines 47-65); a plurality of network-based appliances (one or more peripheral devices)

communicably attached with the web server within the network environment (the host computer communicates with the peripheral devices via communication links such as the Internet) (Yan et al.: column 6, lines 52-67), wherein the network-based appliances each include an embedded device having a non-volatile storage device (as shown in Figure 1, each of the peripheral devices include a virtual machine instruction processor embedded in the peripheral devices) (Yan et al.: column 7, lines 4-8 and 38-45, column 23, lines 53-56 and column 197, lines 14-18); and a loading mechanism (information downloading mechanism) provided on the network-based appliance and operative to download a first application to one of the appliances and a second application to another of the appliances from the web application server upon the occurrence of a power on/off cycle (power on self test which upon boot up time, downloads the functions and applications to the peripheral device; since there can be more than one peripheral device, more than one application can be downloaded to the more than one peripheral device) (Yan et al.: column 20, line 63 - column 21, line 29), wherein a first universal resource locator (URL) is stored on the non-volatile storage device for the one appliance and a URL are stored on the non-volatile storage device for the another appliance, and the first and second application bodies are provided on the web server at a location corresponding with the URL for the one appliance and the another appliance, the URL being initialized to access the respective application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral devices' storage while the server holds the material of the application to be downloaded; since there can be more than one peripheral device that can download information from the web server, more than one URL corresponding to application bodies can be stored on the storage device) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24). This is

further shown in Figures 1 and 2. However, although Yan et al. teach a plurality of network-based appliances and a URL stored on the non-volatile storage device, Yan et al. fail to explicitly teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system for a plurality of network-based appliances taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 2, Yan et al. teach the loading mechanism provided at least in part by the client (the loading of the function and applications occurs in the peripheral device, which is a thin client) (column 19, lines 38-43 and column 21, lines 23-29).

Referring to claim 6, Yan et al. teach each of the application bodies comprises a servlet provided on the web server, as recited in column 5, lines 40-67. According to the definition provided by Hyper Dictionary (<http://www.hyperdictionary.com>), a servlet is “a Java program that runs as part of a network service, typically an HTTP server and responds to requests from clients. The most common use for a servlet is to extend a web server by generating web content dynamically. For example, a client may need information from a database; a servlet can be written that receives the request, gets and processes the data as needed by the client and then returns the result to the client.” Yan et al. teach a network (host computer) that responds to requests from clients. The peripheral devices requests information (applications) from the database and the virtual machines instruction processors process the information and return the results to the peripheral devices.

Referring to claim 7, Yan et al. teach each of the network-based appliances comprising an embedded device and the loading mechanism comprising a virtual machine (the peripheral devices comprises embedded virtual machine instruction processors and the downloading mechanism of the peripheral devices comprises a virtual machine) (column 5, lines 60-67 and column 7, lines 4-8 and further shown in Figure 1).

Referring to claim 8, Yan et al. teach the network-based appliance (peripheral device) using the loading device to download specific appliance configuration settings (column 14, lines 38-40 and column 21, lines 12-17).

Referring to claim 9, Yan et al. teach each of the appliances comprises an embedded device, and the loading mechanism comprising a program routine that copies an application program into the memory of the embedded device from the web server for execution (downloading functions associated with the applications into the peripheral devices) (column 21, lines 23-29 and column 23, lines 53-59).

Referring to claim 10, Yan et al. teach a computer peripheral program product comprising a web application server configured as a central location to store applications and configuration settings for computer peripherals to enable configuration setups for a plurality of computer peripherals (a host computer communicating with the peripheral devices over the network; the host computer stores applications requiring use of peripheral devices and peripheral device profiles for automatic configuration of the peripheral device) (Yan et al.: column 6, lines 52-67, column 20, line 63 - column 21, line 37 and column 23, lines 47-65); a network environment (Yan et al.: column 2, lines 54-60); at least two computer peripherals (a plurality of peripheral devices) (Yan et al.: column 6, lines 63-67 and Figure 1); and an application loader to load a unique extendable architecture application to each of the at least two computer peripherals (downloading applications and functions to the peripheral devices; since there can be more than one peripheral device, more than one application can be downloaded to the more than one peripheral device) so as to enable versioning, updating and remote configuration of the at least two computer peripherals (Yan et al.: column 23, lines 47-50 and column 24, lines 7-9) via the web application server, wherein the application loader associates first and second application bodies of the web application server (the URLs of the applications to be downloaded from the web server can be downloaded to reside in the peripheral devices' storage while the server holds

the material of the applications to be downloaded) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), each of the applications include a respective uniform resource locator (URL) to the application, the application body including one or more individual applications that can be loaded on the computer peripheral (more than one document can be downloaded to the plurality of peripheral devices) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), the URL being initialized to access the application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral device's storage while the server holds the material of the application to be downloaded; since there can be more than one peripheral device that can download information from the web server, more than one URL corresponding to application bodies can be stored on the storage device) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24). This is further shown in Figures 1 and 2. However, Yan et al. fail to explicitly teach an application header having identification information for the application, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header having identification information for the application, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al.

before him at the time the invention was made, to modify the application loading system for a plurality of network-based appliances taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 11, Yan et al. teach each of the computer peripheral comprises a virtual machine including a web client (thin client) (column 7, lines 38-45 and column 19, lines 38-40).

Referring to claim 12, Yan et al. teach one of the computer peripherals (peripheral devices) comprising a printer (as shown by reference character “102B” in Figure 10), and updating comprising configuring the one printer with a printer application comprising a printer configuration state (column 21, lines 12-17 and column 23, lines 58-65).

Referring to claim 13, Yan et al. teach the printer configuration state comprising user settings (user selects a specific operation to perform and the system configures and sets the device) (column 21, lines 12-17).

Referring to claim 14, Yan et al. teach a servlet on the web application server that transfers applications and settings to the printer in response to a power cycle that automatically

updates the application and configuration settings for the printer (column 20, lines 63-67 and column 21, lines 1-5).

Referring to claim 16, Yan et al. teach, in column 5, lines 40-67, column 6, lines 52-67 and column 21, lines 1-29, a method for updating applications to embedded devices (downloading applications to peripheral devices), comprising providing a plurality of network-based appliances each communicably attached with a web application server (a plurality of peripheral devices such as printers, cameras and telephones attached to a host computer via a network) (Yan et al.: column 6, lines 51-67 and Figure 1), each of the appliances having a loading mechanism to download an application to the appliance from the server; updating the appliance with a first separate, dedicated application from the server including a first set of user settings for the one appliance and a second, separate, dedicated application from the server including a second set of user settings unique from the first set of user settings and for another appliance upon the occurrence of a power on/off cycle (the peripheral devices have a virtual machine instruction processor embedded in the devices to facilitate distributed computer between the devices and the host device so that each peripheral device executes power on self test which upon boot up time and downloads the functions and applications to the peripheral device; the downloaded functions and applications include user setting, i.e. specific user selected operations to be perform and configured/set by the system; furthermore, since there can be more than one peripheral device that can download information from the web server, more than one application can be downloaded to more than one appliances) (Yan et al.: column 7, lines 1-5 and 37-48 and column 20, line 63 - column 21, line 29). However, although Yan et al. teach a plurality of appliances communicating with the web server (Yan et al.: Figure 1) and updating

the plurality of appliances with the respective applications from the server upon the occurrence of a power on/off cycle (configuring the peripheral device to download the respective applications concerned with using the functions associated with the peripheral device upon boot up) (Yan et al.: column 20, line 54 – column 21, line 37 and column 23, lines 47-65), Yan et al. fail to explicitly teach querying multiple appliances with the web server to determine presence of an application header for the multiple appliances and updating the plurality of appliances with respective applications upon detecting the presence of the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach querying an appliance with the web server to determine presence of an application header and updating the appliance with the application upon detecting the presence of the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and checking the header, the application from the application server can be downloaded based on the header information in the data file) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system for a plurality of appliances taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the

space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 17, Yan et al. teach the appliance comprising an embedded device, and updating comprising configuring the embedded device with an application (downloading an application) comprising an embedded device configuration state (column 21, lines 12-29).

Referring to claim 18, Yan et al. teach the embedded device configuration state comprising user settings (user selects a specific operation to perform and the system configures and sets the device) (column 21, lines 12-17).

Referring to claim 19, Yan et al. teach a servlet on the web application server that is transferred to the embedded device in response to a power cycle that automatically updates the application and configuration settings for the embedded device (column 20, lines 63-67 and column 21, lines 1-5).

Referring to claim 20, Yan et al. teach a plurality of appliances communicably attached with the web application server (as shown in Figure 1) each with a dedicated one of the loading mechanism, wherein the web application server stores appliance applications and configuration settings (column 20, lines 63-67 and continuing onto column 21, lines 1-29) to enable plural appliance configuration setup to version and update such applications (column 23, lines 47-50 and column 24, lines 7-9).

Referring to claim 21, Yan et al. teach an appliance application loading system for a network environment (Yan et al.: column 2, lines 54-60). Specifically, Yan et al. teach a client (a thin client via the peripheral device) (Yan et al.: column 19, lines 38-40); a server communicating with the client (a host computer communicating with the peripheral devices over

the network) (Yan et al.: column 6, lines 52-67); a plurality of network-based appliances (one or more peripheral device) communicably attached with the web server within the network environment (the host computer communicates with the peripheral devices via communication links such as the Internet (Yan et al.: column 6, lines 52-67), wherein the network-based appliances each including an embedded device having a non-volatile storage device (as shown in Figure 1, each of the peripheral devices include a virtual machine instruction processor embedded in the peripheral devices) (Yan et al.: column 7, lines 4-8 and 38-45, column 23, lines 53-56 and column 197, lines 14-18); and a loading mechanism (information downloading mechanism) provided on the network-based appliance and operative to download a unique application to each of the appliances from the web application server upon the occurrence of a power on/off cycle (power on self test which upon boot up time, downloads the functions and applications to the peripheral device; since there can be more than one peripheral device, more than one application can be downloaded to the more than one peripheral device) (Yan et al.: column 20, lines 63-67 through column 21, lines 1-14 and 23-29), wherein a universal resource locator (URL) is stored on the non-volatile storage device of each appliance, and an application body is provided on the web server at a location corresponding with the URL for each appliance, the URL being initialized to access the application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral devices' storage while the server holds the material of the application to be downloaded; since there can be more than one peripheral device that can download information from the web server, more than one URL corresponding to application bodies can be stored on the storage device) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), wherein servlet settings corresponding to

each of the network-based appliances are automatically updated via the loading mechanism to each appliance if a user locally changes settings of the network based appliance (when one or more of the peripheral device has, upon user initiation, upgraded or updated its settings, the peripheral device automatically sends a notification applet to the host computer notifying the host that the peripheral device has been changed) (Yan et al.: column 23, lines 46-67 through column 24, lines 1-8). However, although Yan et al. teach a plurality of network-based appliances and URLs stored on the non-volatile storage device, Yan et al. fail to explicitly teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system for a plurality of network-based appliances taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise

Art Unit: 2173

been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 22, Yan et al. teach a dedicated servlet configured to have settings that are unique to each of the network-based appliances (each peripheral device of the plurality of peripheral devices registers a peripheral profile with a peripheral database containing a configuration applet that which cause the device to work in a predetermined manner) (column 23, lines 46-67).

3. Claims 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yan et al. U.S. Patent 6,003,065 and Inoue et al. U.S. Patent 6,456,388, as applied to claim 21 above, and Iwase et al. U.S. Publication 2002/0097262.

Referring to claim 23, Yan et al. and Inoue et al. teach all of the limitations as applied to claim 21 above. Specifically, Yan et al. and Inoue et al. teach a first network-based appliance and a second network-based appliance (a plurality of peripheral devices attached to the server via a network) (Yan et al.: column 6, lines 51-67 and Figure 1), wherein each peripheral device has a loading mechanism for downloading applications to the device and each peripheral device is remotely located from one another (the virtual machine instruction processor on each peripheral device facilitates distributed processing on a computer environment) (Yan et al.: column 6, line 51 – column 7, line 8, column 7, lines 37-48 and column 20, line 63 - column 21, line 29).

However, Yan et al. and Inoue et al. fail to explicitly teach the loading mechanism sets up each

network-based appliance to have user setting for a corresponding user who uses the appliance locally. Iwase et al. teach a system that loads information from a server to a peripheral device (transmitting information from a server to a multifunctional peripheral device) (Iwase et al.: page 1, paragraphs 0010-0012) similar to that of Yan et al. and Inoue et al. In addition, Iwase et al. further teach setting up the peripheral device to have user setting for a corresponding user using the peripheral device (transmitting user setting information from a server to a multifunctional peripheral device of the network) (Iwase et al.: page 1, paragraphs 0010-0012). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al., Inoue et al. and Iwase et al. before him at the time the invention was made, to modify the system for loading applications to a connected peripheral device of Yan et al. and Inoue et al. to include the loading of user setting information to the connected peripheral device taught by Iwase et al. One would have been motivated to make such a combination in order to simplify the complicated operation of configuring, or setting a plurality of devices for the user, thereby enhancing the operability of each user.

Referring to claim 24, Yan et al., as modified, teach the one network-based appliance comprises one printer and the another network-based appliance comprises another printer (a plurality of peripheral devices are connected to the host computer via the network, including printers) (Yan et al.: column 22, lines 43-57, column 23, lines 13-46 and Figure 1), the server is configured to update each unique application with software (the peripheral device downloads updates and software upgrades from the server) (Yan et al.: column 23, line 66-column 24, line 27), and the loading mechanism is configured to download a first and second updated unique application to the one printer and the another printer, respectively software (peripheral devices,

i.e. printers, etc. download updates and software upgrades from the host server) (Yan et al.: column 23, line 66-column 24, line 27). Although Yan et al. do not explicitly teach updating each unique application with fixes to software viruses, Yan et al. do teach updating application with software, i.e. updates and software updates, and since fixes to software viruses are a common type of software that can be downloaded from a host, it would have been obvious for one of ordinary skill in the art to update applications with fixes to software viruses. One would have been motivated to make this combination for security purposes, in order to protect their computers or devices from data corrupting viruses.

Referring to claim 25, Yan et al., as modified, teach the loading mechanism utilizes the application body comprising a servlet (Yan et al.: column 5, lines 40-67), wherein the one printer is delivered one servlet and the another printer is delivered another servlet (the peripheral devices request information from the database and the virtual machines instruction processors process the information and return the results to the peripheral devices; for example, a user having a large print job can distribute different portions of a print job to a plurality of printers by querying the database with the applications to determine the available printers and then loading the printer applet and corresponding data into the respective printers) (Yan et al.: column 22, lines 43-57 and column 23, lines 13-46), wherein the one servlet comprises specific appropriate user settings for the one printer and the another servlet comprises specific appropriate user settings for another printer (transmitting user setting information from a server to a multifunctional peripheral device of the network, including a printer) (Iwase et al.: page 1, paragraphs 0010-0012).

4. Applicant's arguments filed 12 January 2005 have been fully considered but they are not persuasive.

5. The examiner respectfully notes that applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Absent the applicant's mere allegation that the cited prior art does not teach or suggest the features of the amended claims, i.e. a plurality of network based appliances with a loading mechanism downloading a respective application to each appliance, the applicant fails to provide arguments specifically pointing out how the claim language differentiate the claims from the prior art. Furthermore, in response to applicant's argument that the prior art does not teach the features of a plurality of network-based appliances, with a loading mechanism downloading a respective application to each appliance, the examiner respectfully disagrees. Yan et al. teach one or more peripheral devices, as recited in column 6, lines 63-67. In addition, Figure 1 shows a plurality of peripheral devices (network-based appliances) connected to the network. Yan et al. further teaches the ability to download applications, such as software updates, etc. to the peripheral devices (column 20, line 63 - column 21, line 29); since there are a plurality of peripheral devices, each of the peripheral devices shown in Figure 1 and can download information from host server. Therefore, the examiner respectfully maintains that the prior art teaches the subject claims.

*Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ting Zhou whose telephone number is (571) 272-4058. The examiner can normally be reached on Monday - Friday 8:30 am - 6:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached at (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-4058.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

11 February 2005



JOHN CABECA  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY